

HTML & CSS basis

Les 2:

Introductie in JavaScript



Onderwerpen les:

- Wat is Javascript?
- Variabelen
- Data types
- Output & input
(Console.log, Alert, Prompt en Confirm)
- Type conversions
- Conditionele logica
- Loops
- Functies

Leerdoelen

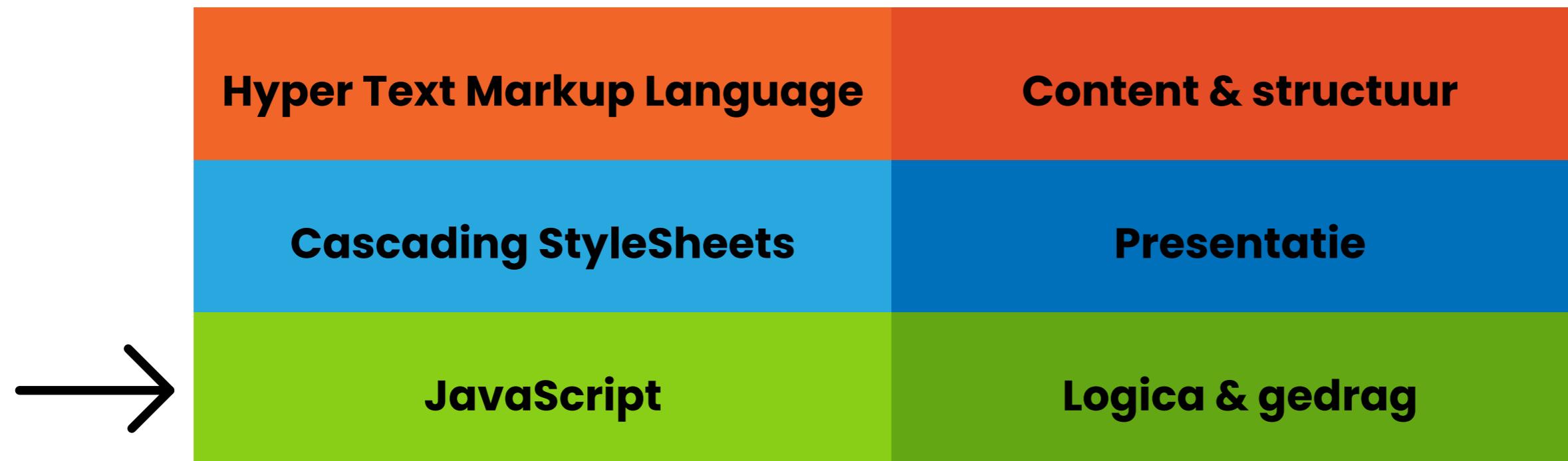
- ✓ Aan het einde van de dag ben je bekend met de basis van Javascript.
- ✓ Aan het einde van de dag kan je uitleggen waarvoor we Javascript gebruiken.
- ✓ Aan het einde van de dag weet je welke datatypes je kunt gebruiken.
- ✓ Aan het einde van de dag kan je informatie plaatsen in een console.log.
- ✓ Aan het einde van de dag kan je een simpele functie aanroepen.
- ✓ Aan het einde van de dag kan je een for loop maken.
- ✓ Aan het einde van de dag kan je conditionele logica toepassen.

JavaScript

Wat is JavaScript?

JavaScript is een veel gebruikte programmeertaal. Wij gaan JavaScript gebruiken om logica en gedrag toe te voegen aan onze webpagina's zodat deze interactief worden.

Javascript kan HTML en CSS updaten en veranderen.



JavaScript

Client side vs server side

Javascript gebruiken wij net als HTML en CSS als een client side script-taal. De code wordt gedownload naar de browser van de bezoeker en daar uitgevoerd.

Een server side script-taal zoals Java, C# en anderen draaien op een andere locatie dan de browser. De front end kan wel deze server aanroepen om gebruik te maken van deze fucn-ties met bijvoorbeeld endpoints.

JavaScript kan ook gebruikt worden als een server side script-taal, maar dat gaan wij niet behandelen tijdens deze fase.

JavaScript

JavaScript chrome dev tools

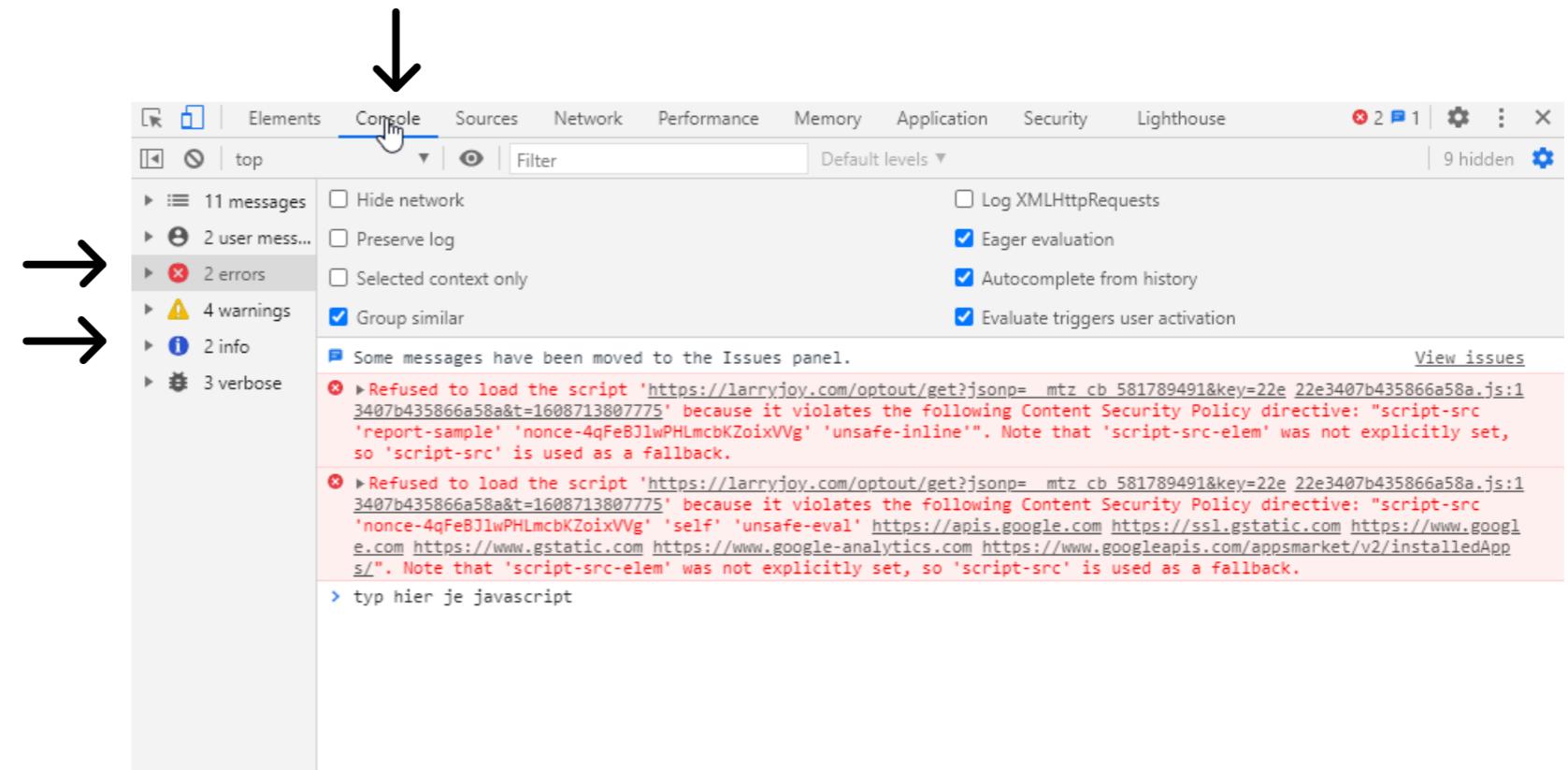
Voor debuggin en leuk om mee te experimenteren!

Controleer HTML CSS en Javascript en haal bugs uit je website.

Je kan errors in javascript zien. Je kan output in de console bekijken en je kan javascript runnen in de console!

Openen DevTools:

- rechter muisklik -> inspecteren of functie f12
- klik op console hier kan je javascript typen
- klik op errors voor debugging
- klik op info om de console.log te zien



Variabelen

var, let, const

JavaScript kent 3 soorten variabelen

var

- heeft geen scope, laatste waarde telt.
- kan opnieuw een waarde krijgen
- kan opnieuw gedeclareerd worden

let

- leeft alleen binnen zijn eigen scope
- kan niet opnieuw gedeclareerd worden binnen dezelfde scope
- kan opnieuw een waarde krijgen

const

- leeft alleen binnen zijn eigen scope
- kan niet opnieuw gedeclareerd worden binnen dezelfde scope
- kan niet opnieuw een waarde krijgen

JavaScript

Verschil JS en Java

strongly vs loosely/weakly typed

In tegenstelling tot Java hoeft je bij JavaScript niet aan te geven welke datatype je in een variabele stopt.

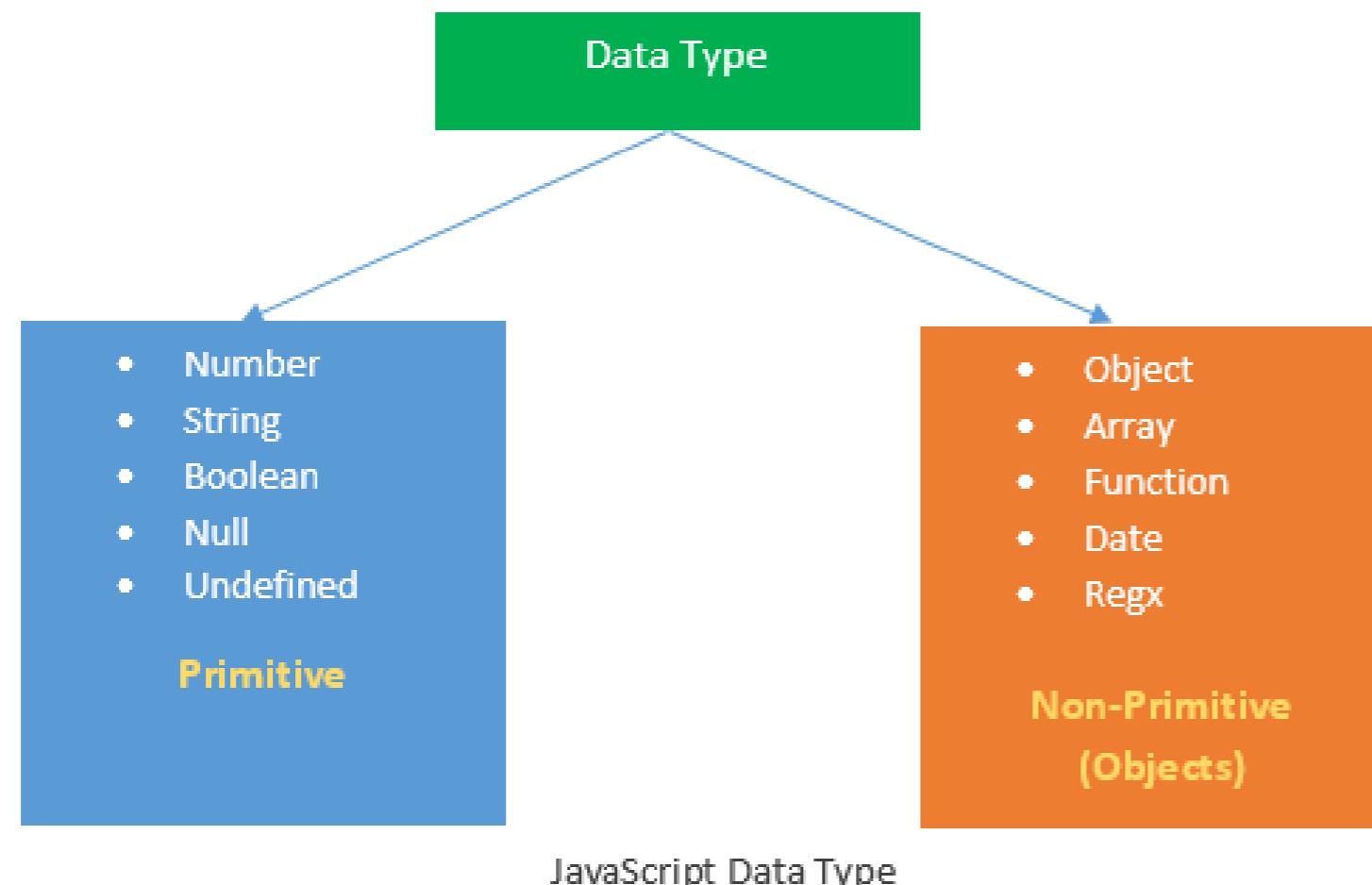
| https://www.w3schools.com/js/js_variables.asp

JavaScript

Datatypes

Number, String etc

Net als Java heeft JavaScript verschillende soorten data types.



Wij gaan ons tijdens deze fase vooral bezig houden met Number, String, Boolean, Array en Function.

Input en output

console.log

Een geweldige tool om te testen! Hiermee kan je iets in de console te loggen. Verwacht je een bepaalde uitkomst van een functie, wil je controleren wat er in een variabele is opgeslagen? log het in de console!

```
console.log('dit is een log' + var1);
```

| https://www.w3schools.com/jsref/met_console_log.asp

Input en output

alert

Deze invoer is onjuist! Het programma is vastgelopen! Uw sessie is verlopen! En vele andere belangrijke meldingen zie je vaak in een alert. In een alert kan je een variabele melden of een String of een combinatie van beiden.

```
alert('dit is een alert + var1);
```

| https://www.w3schools.com/jsref/met_win_alert.asp

Input en output

prompt

Hiermee kan je om invoer van de gebruiker vragen. Deze invoer kan je direct opslaan als een variabele. Een prompt kan je een default waarde geven.

```
var naam1 = prompt("Please enter your name");
```

```
var naam2 = prompt("Please enter your name", "Harry Potter");
```

| https://www.w3schools.com/jsref/met_win_prompt.asp

Input en output

confirm

Een confirm methode geeft een dialoog venster met een bericht, OK knop en een CANCEL knop. De methode returned een true of een false.

```
confirm("Press a button!\nEither OK or Cancel.");
```

| https://www.w3schools.com/jsref/met_win_confirm.asp

Type conversion

Type conversion in JavaScript

Een variabele van een bepaald data type kan worden omgezet naar een ander data type met een JavaScript functie en soms gebeurt het automatisch.

Number()

```
Number("3.14")
Number(Math.PI)
Number(" ")
Number("")
```

```
Number("99 88")
Number("John")
```

dit werkt niet

Type conversion

Type conversion in JavaScript

Unary + operator

```
let y = "5";    // y is a string
let x = + y;    // x is a number
```

Type conversion

Type conversion in JavaScript

String() & toString

```
String(x)      // returns a string from a number variable x  
String(123)    // returns a string from a number literal 123  
String(100 + 23) // returns a string from a number from an expression
```

```
x.toString()  
(123).toString()  
(100 + 23).toString()
```

Type conversion

Type conversion in JavaScript

Meer methodes

Naast de voorbeelden op de vorige pagina's kun je ook booleans omzetten naar numbers of strings, datums naar strings en meer.

| https://www.w3schools.com/js/js_type_conversion.asp

Conditionele logica

if-statement

Een if statement wordt gebruikt om iets uit te voeren als aan een voorwaarde wordt voldaan.

```
var leeftijd = 20;

if (leeftijd > 16) {
    console.log("je bent ouder dan 16");
}
```

| https://www.w3schools.com/js/js_if_else.asp

Conditionele logica

Comparison Operators

Om te controleren of een statement true of false moet returnen gebruiken we comparison operators:

Operator	Description	Comparing	Returns
==	equal to	x == 8	false
		x == 5	true
		x == "5"	true
===	equal value and equal type	x === 5	true
		x === "5"	false
!=	not equal	x != 8	true
!==	not equal value or not equal type	x !== 5	false
		x !== "5"	true
		x !== 8	true
>	greater than	x > 8	false
<	less than	x < 8	true
>=	greater than or equal to	x >= 8	false
<=	less than or equal to	x <= 8	true

Conditionele logica

Logical operators

Met logical operators wordt de logica tussen variabelen of values bepaald.
Wanneer je 18+ bent EN een ticket hebt mag je naar binnen.
Wanneer je student OF docent bent mag je de es bijwonen.

Operator	Description	Example
&&	and	$(x < 10 \&\& y > 1)$ is true
	or	$(x == 5 y == 5)$ is false
!	not	$!(x == y)$ is true

| https://www.w3schools.com/js/js_comparisons.asp

Conditionele logica

if, else if & else

Het kan ook voorkomen dat je meerdere situaties hebt waarbij je verschillende acties wil uitvoeren. Hiervoor gebruiken we if, else if en else

```
var leeftijd = 20;  
if (leeftijd < 16) {  
    console.log("Je bent nog geen 16. Je mag geen alcohol");  
}  
else if (leeftijd >= 16 && leeftijd < 18) {  
    console.log("je bent oud genoeg voor bier. Maar je mag  
    geen sterke alcohol");  
} else {  
    console.log("je bent 18 of ouder en mag bier en sterke alcohol")  
}
```

Loops

for loop

Een for statement is een loop die wordt uitgevoerd zolang een conditie waar is.

```
for (var x = 0; x < 5; x++) {  
    console.log(x);  
}
```

Een for loop bestaat uit 3 statements:

- 1: een teller. Begint meestal bij 0
- 2: De conditie waaraan moet worden voldaan. Bijvoorbeeld x moet kleiner zijn dan 5
- 3: Telt of trekt een getal af van de teller. Wordt uitgevoerd na het code block na elke loop.

In stap 2 kan je bekijken of iets kleiner is dan bv 5. Maar je kan ook controleren of iets groter is dan. of groter of gelijk aan. Dit doen we met comparison operators: < , > , >=, <=

Opdracht

Rock Paper Scissors

Het spelletje 'steen, papier, schaar' is een leuke manier om de geleerde technieken toe te passen! Steen verslaat schaar, schaar verslaat papier en papier verslaat steen. Kies jij hetzelfde als je tegenstander dan is het gelijkspel.

Stap 1: Steen papier of schaar

Maak 3 buttons/tekst elementen die een functie aanroepen. Deze functie geeft voor nu alleen een alert met steen, papier of schaar.

Lukt het jou om slechts 1 functie te maken en een argument mee te geven?

Stap 2: Zorg voor een random keuze van de PC

Maak een array (zie zelfstudie) met de 3 keuzes.

Zorg voor een random keuze uit de array met Math.floor (zie zelfstudie)

Maak een simpele alert

Roep de functie aan nadat de speler een keuze heeft gemaakt.

Extra

Conditionele logica:

| https://www.w3schools.com/js/js_switch.asp

Loops:

| https://www.w3schools.com/js/js_loop_while.asp

| https://www.w3schools.com/jsref/jsref_foreach.asp

Array:

| https://www.w3schools.com/js/js_arrays.asp

Random:

| https://www.w3schools.com/js/js_random.asp

Loops

for loop

Met een for loop kunnen we gemakkelijk loopen over een array.

```
let boodschappenLijst = ["jam", "kaas", "melk", "boter", "appel"];
var deString = "";
for (var x = 0; x < boodschappenLijst.length; x++) {
    deString += boodschappenLijst[x] + "<br>";
}
```

.length -> hiermaar krijg je de lengte van de array. In het voorbeeld is dit 5.

< -> betekent kleiner dan. Zie lijst comparisons javascript. (==, ===, !=, <. >, >=, <=)

| https://www.w3schools.com/js/js_loop_for.asp

Functions

function in JavaScript

Een blok met code die een bepaalde taak / taken uitvoert zodra deze wordt aangeroepen. Het gene dat de functie aanroept kan informatie meegeven dat de functie vervolgens kan gebruiken.

```
function myFunction() {  
    alert("iets uitvoeren");  
}
```

```
function myFunction(paramter1, p2) {  
    alert(parameter1);  
    console.log(p1);  
}
```

Functions

Aanroepen functie

Een functie kan worden aangeroepen vanuit JavaScript.

```
myFunction(12,5)
```

Een functie kan worden aangeroepen vanuit een event. Een event is een gebeurtenis op de website. Dit kan een klik op een knop zijn, een wijziging in een invoer veld van een formulier of het drukken op enter. Het binden van een event en een functie kun je in je javascript koppelen aan een HTML element ook kun je in je HTML dit event al koppelen aan een HTML element. zie voorbeeld:

```
<button onclick="myfunction()">Click me</button>
```

Functions

Return

Een functie kan ook iets teruggeven via een return statement.

```
function squaremeter(length, width) {  
    sm = length * width;  
    return sm;  
}
```

```
console.log(squaremeter(3,5);
```

- | https://www.w3schools.com/js/js_functions.asp
- | https://www.w3schools.com/js/js_events.asp

Manipuleren HTML

Ophalen HTML element

Als je de inhoud van een HTML element wil veranderen, of misschien de style zoals de kleur of wil swtichen tussen display block en display none dan moet Javascript wel weten welk element je wil veranderen. Het element in het HTML document heeft een tag soms een id en soms een class. Hiermee ga je het element ophalen.

```
var x = document.getElementById('naamid');
```

ophalen 1 HTML element uit document

```
var y = document.getElementsByClassName('naamclass');
```

ophalen 1 of meerdere HTML element uit document. Dit wordt dan een array

Manipuleren HTML

Inhoud HTML element veranderen met innerHTML

De innerHTML is de content tussen de openingstag en de sluitingstag. <div>innerHTML</div>
De inhoud hiervan kan je overschrijven met javascript.

```
x.innerHTML = "nieuwe inhoud";
```

style HTML element veranderen met .style

De CSS styling van een element kan je veranderen met javascript

```
x.style.color = "blue";
```

Opdracht

Stap 3: Vergelijk de waarden met elkaar

De waarde van de speler heb je opgehaald in stap 1, deze vergelijk je met de waarde van de pc die gereturnd wordt vanuit stap 2.
Met if statements of switches (zie zelfstudie) bekijk je wie de winnaar is: Speler of PC.
Geef een alert you lose of you win.

Stap 4: HTML ipv alerts

Ipv alerts plaats je tekst in de HTML.
Speler kiest: steen
PC kiest: papier
PC wint!

Extra

Stap 5: Geef extra informatie

Steen is hetzelfde als Steen. Gelijk spel!

Steen wordt bedekt door Papier. PC wint!

Papier wordt geknipt door Schaar. Jij wint!

etc

Stap 6: Make it pretty

Heb je al wat kennis van CSS? Maak het spelletje mooi. (wacht anders tot de volgende les)

JavaScript

Linken naar javascript

Het HTML bestand moet toegang hebben tot de code in het JS bestand. Voor het einde van de body plaats je het volgende:

```
<script src="myscripts.js"></script>
```